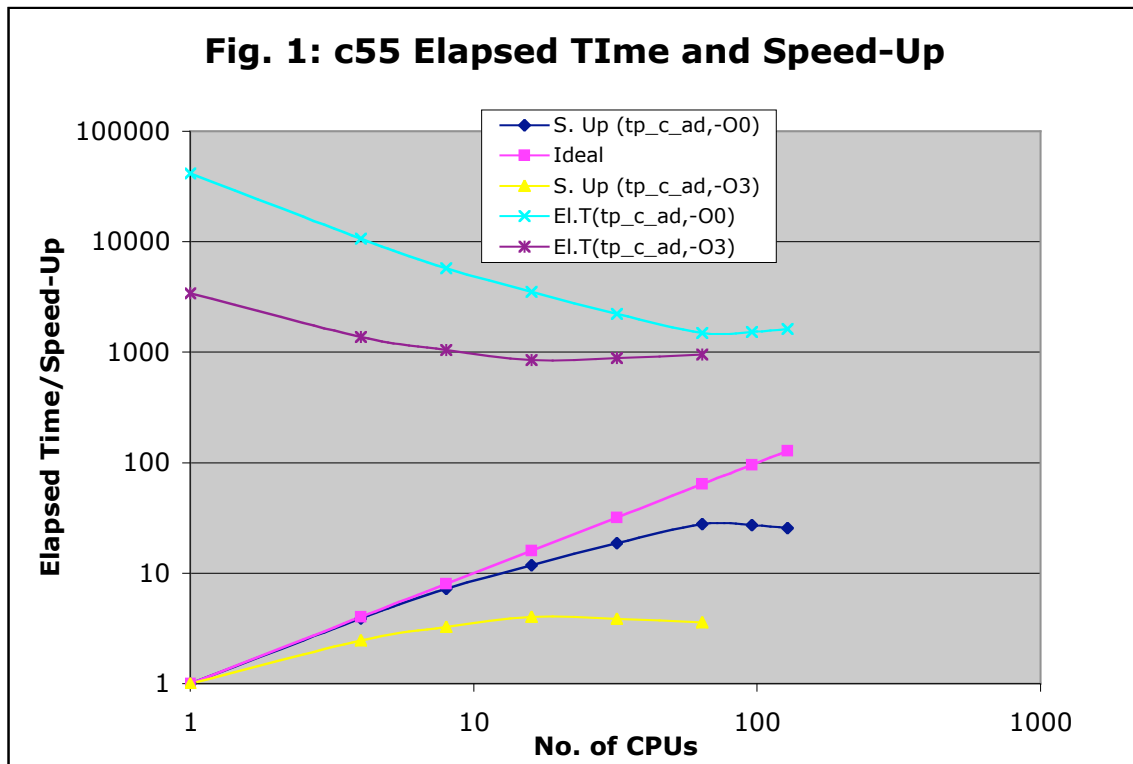


Adjoint work: Second Status Report

1. The code was built on the Altix with OpenMP enabled. The first port to the Altix that gave correct results, that is, with `tp_core_ad.F90` compiled at `-O0` is considered the baseline. Figure 1 shows elapsed time (ET) and speed-up (SU) versus number of CPUs for the baseline. The figure also shows ET and SU for the “compiler-optimized” build, that is, with `tp_core_ad.F90` compiled at `-O3` and dummy aliasing enabled. The former scales “only fairly” up to 64 CPUs (with SU of about 28) while the latter will not scale beyond 16 CPUs (with SU of about 4.0). Looking at the elapsed times, the combination of OpenMP with “compiler optimization” for `tp_core_ad.F90` reduced the elapsed time from about 41400 seconds for two time steps and five test vectors on a single CPU (with OpenMP disabled and `-O0` for `tp_core_ad`) to about 850 seconds on 16 CPUs (with OpenMP enabled and `-O3` for `tp_core_ad`), that is an aggregate speed-up of about 49. In summary, we started at about 7.76 hours per time step for five test vectors for the c55 case and we are down to about 7.1 minutes.



2. All attempts to manually recode the adjoint code to avoid dummy argument aliasing have not produced the desired results. In the meantime, this will be suspended so that attention can be focused on improving sequential performance and scalability.
3. We have started looking for opportunities for sequential tuning and we will report progress in due course.

4.